LANGLEY GRANT
IN-61-CR
110420
8P

FINAL TECHNICAL REPORT

METHODS FOR DESIGN AND EVALUATION OF INTEGRATED HARDWARE/SOFTWARE SYSTEMS FOR CONCURRENT COMPUTATION

NASA Research Grant NAG-1-467

Grant Period: March 15, 1984 - September 30, 1987

Principal Investigator:
Terrence W. Pratt, Research Professor

Department of Computer Science
University of Virginia
Charlottesville, VA 22901

December 1987

Project Overview

The research project supported by this grant has become known informally as the PISCES Project (an acronym for *Parallel Implementation of Scientific Computing EnvironmentS).* The long term objective of the PISCES project is to develop better ways to evaluate the new parallel computer systems that are coming on the market. The particular target is evaluation of the effectiveness of these parallel systems for the large- scale scientific software systems of interest to NASA. The ultimate goal of the work is to influence the design of the next generation of parallel systems to better meet the needs of NASA software systems.

The major problem areas in evaluating the emerging commercial parallel machines lie in several facts. First, the large variety of different parallel architectures available makes evaluation difficult. To reprogram several large NASA systems for each different architecture in order to evaluate performance differences would be expensive and time-consuming.

Second, evaluation of these machines should be in the context of the same sort of programming environments, using Fortran or similar high-level languages, that have traditionally been available on sequential machines. These software layers have an impact on performance that is important to evaluate.

Third, the performance of parallel machines on large NASA software systems is the most important basis for evaluation. Performance on small test programs or on particular algorithms is not of as much interest.

Fourth, the evaluation must take into account the amount of reprogramming of the existing NASA software base that would be required to effectively use these parallel machines. Most of this existing code is in Fortran.

Approach

The PISCES project approach is to build a series of "testbed" programming environments to support the evaluation of a large range of parallel architectures. The conceptual basis of the PISCES environments lies in a focus on careful definition of the underlying "virtual machine" provided by the PISCES system for the user. The goal is to implement the same virtual machine on each target architecture, by porting the PISCES software to each machine.

To experiment with Fortran extensions for expressing parallelism, the PISCES environments are designed so that new Fortran extensions can be easily implemented. To experiment with the porting of existing NASA Fortran-based codes, the PISCES environments are designed to allow Fortran codes to be run on various parallel architectures with minimal change. The PISCES environments provide support for automatic collection of performance data, so that performance of large systems may be easily monitored and measured on different architectures.

Major Results

Two PISCES environments have been designed and implemented during this grant. The PISCES 1 environment was implemented during 1984-85 on a DEC VAX uniprocessor under the UNIX operating system. PISCES 1 was later ported to a network of Apollo workstations. The PISCES 2 environment was implemented during 1986-87 for the NASA Flexible FLEX/32 parallel computer. Work on the PISCES 2 system and its applications is continuing through the Institute for Computer Applications in Science and Engineering (ICASE) at NASA Langley Research Center. A new formal model of parallel computation has been developed as part of the Ph.D. research of P.D. Stotts.

Summary of Research Activities

The major tasks performed during this research grant have been the following.

The PISCES 1 Parallel Programming Environment

The first major phase involved the design and implementation of the PISCES 1 parallel programming environment [2,7]. The major parts of this activity were the following:

1. *Design of the PISCES 1 environment.* Building on some design work at ICASE that preceded the grant period, we successfully completed the design of a PISCES 1 parallel programming environment, based on Fortran 77 and UNIX. The environment provides parallelism to the programmer at two different "granularities" of operation: "clusters" and "tasks". The parallel "virtual machine" defined by the PISCES software was carefully specified in the User's Manual for the system [2]. The virtual machine was made visible to the programmer as a set of extensions to Fortran 77 for controlling the operation of the virtual machine (the run-time environment for program execution).

2. *Implementation of PISCES 1 on a uniprocessor.* With the help of two University of Virginia graduate students, Nancy Fitzgerald and Jeff Taylor, who spent Summer 1984 at NASA Langley, a prototype implementation of PISCES 1 was completed for the DEC VAX under the UNIX operating system. This uniprocessor implementation provides simulated parallelism by using UNIX processes, which time-share the CPU. The programmer writes programs in Pisces Fortran (Fortran 77 with parallel extensions). Pisces Fortran programs are translated by a preprocessor into standard Fortran 77. A run-time library implements the various parts of the PISCES 1 virtual machine on the VAX. A complete record of the initiation and communication of parallel tasks is left on a set of files after a run. A set of analysis programs may then be used to collect summary statistics on program performance [3]. The PISCES 1 system was made available to users at the University of Virginia and ICASE. Several substantial applications were programmed for the system (described below).

3. *Apollo workstation network implementation of the PISCES 1 environment.* The PISCES 1 system was successfully ported to a local network of Apollo workstations at the University of Virginia in 1985 by Nancy Fitzgerald. The network consisted of 10 Apollo DN 300

workstations connected by an Ethernet-like local network. The system implements the same PISCES 1 virtual machine as the VAX implementation, so that applications programs may run on either system without change in the code. The programmer sees the workstation network as a single clustered parallel machine. The programmer may distribute parallel applications tasks to these clusters by using Pisces Fortran commands.

In constructing this implementation no attempt was made to optimize performance; we used the same basic implementation strategy as on the VAX. After implementation was complete, Fitzgerald studied the basic performance of the system in her M.S. thesis [5]. The basic performance parameters are the time to initiate a task and the time to send a message. Due to the amount of disk access involved in both activities on the Apollos, both task initiation and message passing were found to be unacceptable slow for realistic applications. To improve this performance we would have to change the operating system on the Apollos, which was not possible at the time. As a result, we did not pursue this implementation.

4. *Applications of the PISCES 1 environment.* Two applications of the PISCES 1 to problems of interest to NASA were the following. Neither was supported directly by this grant, but both illustrate the potentially broad application of the PISCES design.

(a) *Sparse matrix iterative equation solver.* Prof. Merrell Patrick of Duke University, in a collaborative effort with this project, designed a series of six variations on a general parallel solution package for sparse matrix representations of linear systems of equations. These six variations were implemented in Pisces Fortran and were run on the VAX implementation of PISCES 1 at ICASE. The variations included a data flow version, two chaotic versions, and various forms of communication (direct, broadcast, etc.) between the parallel tasks. This work is reported in [4,9].

(b) *Dynamic scene analysis system.* Prof. Worthy Martin and Ph.D. candidate Chew-lim Tan of the U.Va. Computer Science Department used the PISCES 1 system for an application of artificial intelligence techniques to an image processing problem: the tracking of moving objects in dynamically changing scenes. Parallel tasks are used to locate and track different objects within the scene, which is continuously changing as new inputs arrive from the camera. This work has led to a Ph.D. thesis, completed by Tan in May 1986.

The PISCES 2 Parallel Programming Environment

The second major phase of the PISCES project involved the design and implementation of a second parallel programming environment, this time for the NASA Flexible FLEX/32 parallel computer [13,14]. The major parts of this actvity were the following:

1. *Design of the PISCES 2 environment.* The PISCES 2 system provides a rich environment for experimenting with parallel programming concepts. The main concepts are:

(a) The virtual machine is relatively independent of the underlying architecture, so that programs in Pisces Fortran are not written using constructs peculiar to one parallel architecture.

(b) The virtual machine provides several granularities of parallel operation. In PISCES 2 parallel operation is provided between large-grain "clusters of tasks" and "tasks within a cluster" and medium-grain loop iterations and arbitrary program segments.

(c) The virtual machine architecture is "clustered", with each cluster representing a group of processing and memory resources of the underlying machine. The cluster organization is static within a single run, but it may be varied between runs.

(d) The operating system is represented as a static set of tasks that run within each cluster of the virtual machine. User tasks invoke operating system functions by sending messages to these operating system "controller" tasks.

(e) User programs are represented at the top-level by a set of tasks that are dynamically initiated and terminated during a run. Tasks communicate using asynchronous message passing. Message passing queues are infinite (to available memory), and the receiving task may "accept" a message in a variety of different ways, or never.

(f) Many of the "FORCE" constructs developed by Harry Jordan of the University of Colorado are included in PISCES Fortran. These constructs provide medium-grain parallelism, including parallel execution of loop iterations, subroutine calls, and arbitrary program segments. Synchronization mechanisms include barriers and critical regions with lock variables. Communication is via shared variables in COMMON blocks.

(g) The programmer controls the mapping of hardware resources to the "clusters" of the PISCES 2 virtual machine. Each run of a program may be configured differently. Currently the mapping consists of assignment of a group of processors to each cluster for running tasks and "forces".

2. *Implementation of PISCES 2 on the NASA FLEX/32.* Implementation of the PISCES 2 design was completed in June 1987. The PISCES 2 system as implemented on the NASA FLEX/32 consists of several software components:

(a) *Preprocessor.* The extensions to Fortran 77 that form the Pisces Fortran language are implemented with a preprocessor that converts Pisces Fortran to standard Fortran 77, which is then compiled with the FLEX Fortran compiler. The preprocessor converts many parallel constructs to in-line Fortran code. The core parallel constructs are implemented with a small runtime library of routines called from the Fortran code.

(b) *Configuration environment.* Before each individual run of a parallel program, the user works in the PISCES "configuration environment" to choose the configuration options for that particular run. These options include the choice of the number of clusters and the cluster numbering, choice of the mapping of hardware resources to these clusters, creation of loadfiles for downloading to hardware PE's, and choice of run-time options such as time limits, tracing options, etc. These options are chosen with a series of menus and prompts. The user does not need to know FLEX operating system commands for this activity.

Configurations may be saved in files for later editing/reuse in other runs of the same or different programs. Thus a user can build a library of configurations for use in comparative performance evaluation studies.

(c) *Execution environment.* When the user has finished choosing the configuration, execution of a parallel run may be initiated from the configuration environment. When the program begins execution on the FLEX parallel processors, a PISCES "execution environment" provides a menu of commands that allow the user to interact in real-time with the running program. Options include the ability to initiate or kill running tasks, send messages to tasks, monitor execution in real-time, save trace data about key events in trace files, and view the system state (including detailed information about storage management and message queue contents.)

(d) *Postprocessing of trace files.* Trace information saved during a run may be processed off-line after the run to obtain detailed information for timing and debugging parallel program components.

The PISCES 2 system is available on the CSM FLEX/32 to NASA users through the LaRC network. A short course for users of the system was given in April 1987 at LaRC, ICASE, and the University of Virginia.

New Formal Model of Parallel Computation

A new formal model of concurrent computation has been developed by P.D. Stotts in a Ph.D. thesis [8]. The model is based on the mathematical system known as "H-graph semantics" (developed by the principal investigator) together with a "timed Petri net" model of the parallel aspects of a system. Stotts has shown that the model can be used for timing analyses of hierarchical parallel systems and for automatic detection and resolution of conflicts in concurrent access to shared data. We hope to use an extension of this model as the basis for a formal definition of the PISCES 2 virtual machine, which would be potentially of value to the PISCES implementor and user. A paper on this work was presented at a workshop in Europe in summer 1985 [6]; a journal paper has been accepted for publication [15].

Continuing Activities

Although the University of Virginia grant has terminated, the PISCES project is continuing under NASA sponsorship as an ICASE project. The major short-term continuing activities are the following:

1. *Performance measurement and performance tuning of the PISCES 2 system.* The PISCES 2 system has been fully implemented on the FLEX/32, but there is still much to be learned about the performance of this software system for typical parallel algorithms of interest to NASA. Initial comparative measurements of performance by Prof. Merrell Patrick and Mark Jones of Duke University on a Choleski factorization algorithm indicate that PISCES 2 provides performance that is superior to that provided by the Concurrent Fortran system provided by Flexible Corp. for their own FLEX/32 machine. More extensive studies of PISCES 2 performance are currently underway. After we understand where the performance bottlenecks of the system occur, additional "tuning" of the system to improve performance is anticipated.

2. *Implementation of NICE/SPAR as a PISCES 2 program.* The CSM NICE/SPAR testbed system for structural analysis is being modified for parallel operation under PISCES 2. The SPAR modules will be implemented as Pisces Fortran "tasks". The command stream and data base actions processed by NICE will be modified to allow parallel operation. Only minimal change to the system will be made, on the order of what might reasonably be expected in moving any large software system into a parallel environment. The performance speedup of the parallel version of NICE/SPAR will be measured after implementation. With this system we hope to better understand the problems of moving large NASA Fortran-based software systems into a parallel environment, and what performance gains might realistically be expected.

3. *Implementation of new parallel algorithms*. Two existing PISCES 1 applications programs, the sparse matrix package of Patrick and the dynamic scene analysis package of Martin and Tan, are being adapted for PISCES 2 on the FLEX/32. PISCES 2 provides a rich environment for experimentation with new parallel algorithms, and we plan to implement a number of additional new algorithms, working in collaboration with other NASA, ICASE, and university researchers.

4. *PISCES 3 for the NCUBE and Intel hypercubes*. A new parallel processing research center has been established at the University of Virginia, called the Virginia Institute for Parallel Computation (VIPC). Its initial focus will be on software systems and applications for hypercube architecture parallel computers. VIPC is installing large-scale hypercube parallel machines from NCUBE and Intel. The PISCES 2 system is being modified and extended for these hypercube machines. The modified system is named PISCES 3. When available in 1988, PISCES 3 will provide a third environment for experimentation and evaluation of parallel architecures. This work is being funded by non-NASA sources.

Conclusions

The PISCES project is an open-ended project that is expected to continue for several more years. Excellent progress was made during the period of this grant, which funded the initial phases of the project. The PISCES software developed under this grant and installed on the NASA FLEX/32 provides a rich base for further development, experimentation, and evaluation.

Personnel and Degrees Granted

P. David Stotts (Ph.D., 1985)
Nancy Fitzgerald (M.S., 1985)
Jeff Taylor (M.C.S., 1984)
Robert Wise (M.S. expected, 1988)
Mary Jo Haught (M.S. expected, 1988)
Cathy Dean (staff support)

Publications and Presentations

[1] Pratt, T. "The PISCES 1 Parallel Programming Environment", presentation at the *Los Alamos Workshop on Parallel Programming Environments,* August 7-9, 1984, Los Alamos, NM.

[2] Pratt, T. *PISCES 1 User's Manual,* technical report, Computer Science Dept., U. of Virginia, August 1984, 18 pp.

[3] Taylor, J.A. "Performance Analyzer for the PISCES System", M.C.S. Report, Computer Science Dept., U. of Virginia, December 1984, 13 pp.

[4] Patrick, M. and T. Pratt, "Communications Oriented Programming of Parallel Iterative Solutions of Sparse Linear Systems", presented at *Workshop on the Impact of Parallel Architectures on Numerical Methods,* U. of Texas at Austin, March 1985.

[5] Fitzgerald, N.J. *Implementation of a Parallel Programming Environment,* M.S. thesis, Computer Science Dept., Univ. of Virginia, May 1985, 80 pp.

[6] Stotts, P.D. and T.W. Pratt, "Hierarchical Modeling of Software Systems with Timed Petri Nets," International Workshop on Timed Petri Nets, Torino Italy, July 1985. (Also Report 85-01, Computer Science Dept., Univ. of Virginia, January 1985)

[7] Pratt, T. "PISCES: An Environment for Parallel Scientific Computation," *IEEE Software,* 2, 4, July 1985. (Also ICASE Report 85-12, February 1985.)

[8] Stotts, P.D. *A Hierarchical Graph Model of Concurrent Real-time Software Systems,* Ph.D. dissertation, Computer Science Dept., Univ. of Virginia, August 1985, 174 pp.

[9] Patrick, M. and T. Pratt, "Communications Oriented Programming of Parallel Iterative Solutions of Sparse Linear Systems," *Communications in Applied Numerical Methods,* vol. 2, 255-261, 1986.

[10] Pratt, T. "Finding the Right Virtual Machine for Parallel Applications Programming," presentation at *Workshop on Performance Efficient Parallel Programming* (September 1986, Seven Springs, PA, sponsored by NSF and Carnegie-Mellon University).

[11] Pratt, T. "The PISCES 2 Parallel Programming Environment," presentation at *Parallel Languages and Environments Workshop* (November 1986, Williamsburg, VA, sponsored by ICASE and NASA).

[12] Pratt, T. "Short Course: Introduction to the PISCES 2 Parallel Programming Environment," presented at NASA LaRC/ICASE (March 1987) and U. of Virginia (April 1987).

[13] Pratt, T. *PISCES 2 User's Manual (Version 2),* ICASE Interim Report 2, July 1987.

[14] Pratt, T. "The PISCES 2 Parallel Programming Environment," *Proceedings 1987 International Conference on Parallel* Processing, St. Charles, IL, August 1987, 439-445 (also ICASE Report 87-38, July 1987).

[15] Stotts, P.D. and T. Pratt "Petri Net Reachability Trees for Concurrent Execution Rules," *J. of Parallel and Distributed Computing,* (to appear).

[16] Pratt, T. "Methods for Design and Evaluation of Parallel Computing Systems," presentation at *NASA Workshop on Computational Structural Mechanics,* NASA LaRC, November 1987.